The Horcruxes

PPA – CW2 Text based adventure game

Muhammad Chikhoun (23013604)
12-6-2023

The Horcruxes

Introduction

My coursework is based in the world of Harry Potter. For those who aren't aware, Harry Potter is one of the most successful novel series in history and the story was retold through movies. In the story the protagonist, Harry, defeats the evil wizard Voldemort by destroying the various parts of his soul, which are hidden in the form of objects, known as horcruxes. The concept of my game will follow the same idea.

You, the player, will venture through the scenery looking for objects and completing objectives that destroy horcruxes. You will either use items to destroy the horcruxes or give the items to characters who destroy them for you replicating the story in the book.

Tasks

Base Tasks:

- The base code provided the required classes and methods to create the rooms. In order to create the additional rooms, I initiated more objects of the class Room.
- I created a class called Items. One of its attributes is a Boolean, "canPickup". If this value is true, then the item can be picked up, if false, then the item cannot be picked up. The objects of class item also have another attribute, currentRoom, which is of type Room and describes which room the item is located it. This allows the item to be placed at the start of the game, and moved throughout the game as well.
- I also created a player class which has two key attributes that allowed me to complete this task. The first attribute is a maxWeight, which is the maximum weight that the player can hold. Each item has the attribute weight and if It exceeds the limit, the player cannot pick up that item. Once an item is picked up, it is added to the second attribute which is an array list called inventory which holds all the items the character is holding. Once this happens, the item itself also has indication that it is inside the inventory in another Boolean attribute, inInventory.
- In order to implement a back command, I tracked the players movement through the room in a linked list which is used as an attribute of the class player. And then use a function that gets the last room added to the linked list and uses it to take the character back.
- I added numerous commands:
    - Inventory – uses a loop through the array that contains all items in the players inventory and prints it. Also informs the user how much weight is available to carry
    - Use – takes two items when this command is called. It uses the first item mentioned on the second item through if statements that validate the input. The if statements also use loops through items
    - Give – takes one item and one character name when this command is called. The character name is taken first and then the item name. Both are compared to an array list of all items and characters to see if it is a valid input through a loop and if statements.
    - Pickup – This command allows the player to pick up and item and it is added to the array list inventory. The weight of the item is added to the total. Again, a loop through all items and if statements validates the command input.

- Drop – this command allows the user to drop items. If the users weight is too high to pick up an item they want, this allows them to drop other items in the inventory making space for the desired item. This will remove the item from the inventory array list and place it in the room where the player currently is.

Challenge tasks:

- I created a character class, which holds an attribute that is an array list. The array list holds the rooms the character is allowed to walk between, and a random number generator is used to generate which of the neighbouring rooms the character will be in.
- I added extra functions to the parser to allow it to identify three word commands. Similar the way two word commands were implemented
- For the transporter room, I created an array list with all the rooms and used a random number generator to send the player to any random room besides the teleporter room itself by calling the "get" function of the array list class.

Implementation

Coupling:

One example of my use of coupling is reducing dependency on external libraries. The only external libraries used were array lists, and linked lists

Cohesion:

One example of my use of cohesion is grouping related functions. In a class functions or methods that are closely related and work towards a common goal should be grouped together. In the case of my game, I organised all the functions related to the game's response to the users input, together in the game class.

Responsibility-Driven Design:

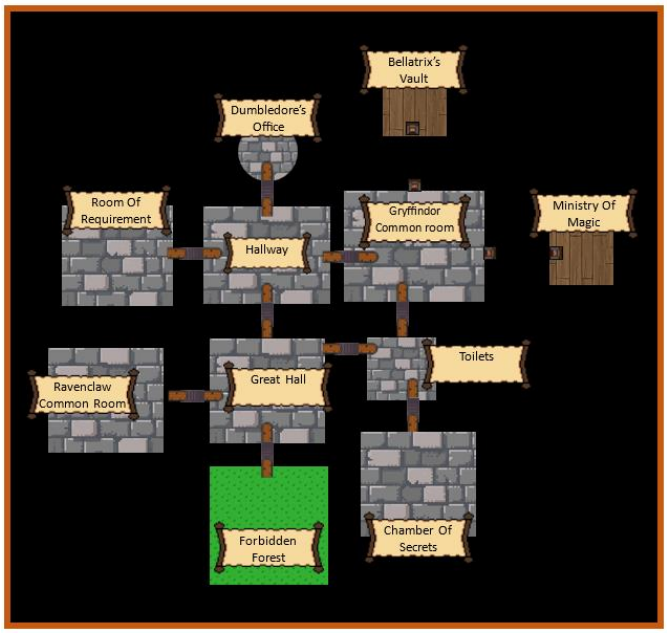
One example of my use of responsibility-driven design is designing classes with clear responsibilities. Each class I created has a clear responsibility. For example, I have classes dedicated for items, characters, rooms, and objectives.

Maintainability:

One example of my use of code maintainability is descriptive variable and method naming. This makes my code easy to read and understand as well as modify and maintain. Additionally, the use of comments further exemplifies its readability and maintainability.

Walkthrough

In order to complete the game the quickest way, use the following walkthrough and map.



Input	Details
Go south	Go to the hallway
Go south	Go to the great hall
Go west	Go to the Ravenclaw common room
Pickup diadem	Pick up the diadem
Back	Go back into the great hall
Go east	Go to the toilets
Go south	Go to the chamber of secrets
Pickup fang	Pick up the fang
Back	Go back to the toilets
Go north	Go north to the Gryffindor common room
Go east	Go to the ministry of magic
Pickup locket	Pick up the locket
Back	Go back to the Gryffindor common room
Go north	Go north to Bellatrix's vault
Pickup cup	Pick up the cup
Back	Go back to the Gryffindor common room
Give ron locket	Give Ron the locket (Ron moves between the toilets and the Gryffindor common room so ensure he is in the room with you)
Use fang diadem	Destroy the diadem using the fang
Pickup diary	Pick up the diary (ensure you are in the Gryffindor common room at this point
Use fang diary	Destroy the diary using the fang
Go west	Go west into the hallway
Give hermione cup	Give the cup to Hermione (Hermione moves between the hallway and the great hall so ensure she is in the room with you)
Go north	Go north into Dumbledore's office
Give dumbledore fang	Give dumbledore the fang
Go south	Go south into the hallway
Go east	Go east into the Gryffindor common room
Pickup sword	Pick up the sword
Back	Go back into the hallway
Give neville sword	Give the sword to Neville (Neville moves between the hallway and the great hall so ensure he is in the room with you)
Go south	Go south into the great hall
Go south	Go south into the forbidden forest

The game should end, now that all the objectives have been completed. Use "help" during the game to see the list of commands available or the remaining objectives to complete. You can also use "inventory" to see what items you are holding and how much space is available, and can use "drop" to remove an item from your inventory to make space for other items if need be. You can also go into the room of requirement to be teleported to a random room.